

RRRRRRRRRRRR		UUU		UUU	NNN		NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN		NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN		NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF	FFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	000000000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	000000000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	000000000	000	FFF	FFF

```
RRRRRRRR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RRRRRRRR  MM      MM
RRRRRRRR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RRRRRRRR  MM      MM
RR      RR  SS      DD      DD  EE      RR      RR  MMMM  MMMM
RR      RR  SS      DD      DD  EE      RR      RR  MMMM  MMMM
RR      RR  SS      DD      DD  EE      RR      RR  MM  MM  MM
RRRRRRRR      SSSSSS      DD      DD  EEEEEEE  RRRRRRRR  MM      MM
RRRRRRRR      SSSSSS      DD      DD  EEEEEEE  RRRRRRRR  MM      MM
RR      RR      SS      DD      DD  EE      RR      RR  MM      MM
RR      RR      SS      DD      DD  EE      RR      RR  MM      MM
RR      RR      SS      DD      DD  EE      RR      RR  MM      MM
RR      RR      SS      DD      DD  EE      RR      RR  MM      MM
RR      RR  SSSSSSSS  DDDDDDDD  EEEEEEEEE  RR      RR  MM      MM
RR      RR  SSSSSSSS  DDDDDDDD  EEEEEEEEE  RR      RR  MM      MM
```

```
LL      IIIII  SSSSSSSS
LL      IIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIII  SSSSSSSS
LLLLLLLLLL  IIIII  SSSSSSSS
```

```
1 0001 0 %TITLE 'Misc. error checks and messages.'
2 0002 0 MODULE RSDERM (
3 0003 0 IDENT = 'V04-000'
4 P 0004 0 %BLISS32[
5 P 0005 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)
6 0006 0 ]
7 0007 0 ) =
8 0008 1 BEGIN
9 0009 1
10 0010 1 *****
11 0011 1 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *
31 0031 1 *****
32 0032 1
33 0033 1 ++
34 0034 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
35 0035 1
36 0036 1 ABSTRACT: Residual error messages. Issues error messages for various
37 0037 1 error conditions for which no error messages have been
38 0038 1 issued.
39 0039 1
40 0040 1 Also contains error detection and error message routines
41 0041 1 that are called from more than one location.
42 0042 1
43 0043 1 ENVIRONMENT: Transportable
44 0044 1
45 0045 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
46 0046 1
```

RSDERM  
V04-000

Misc. error checks and messages.  
Revision History

F 11  
16-Sep-1984 01:40:15  
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]RSDERM.BLI;1

Page 2  
(2)

```

: 48
: 49
: 50
: 51
: 52
: 53
: 54
: 55
: 56
: 57
: 58
: 59
: 60
: 61
: 62
: 63
: 64

0047 1 %SBTTL 'Revision History'
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1 006 REM00006 Ray Marshall 22-June-1983
0052 1 Modified routine TSTTFE to now look to see if we are within a
0053 1 .LITERAL. If so, it issues an error and closes forces a
0054 1 .END LITERAL.
0055 1
0056 1
0057 1 005 KFA00005 Ken Alden 18-Mar-1983
0058 1 Made SAVE/RESTORE related items visible to DSR.
0059 1
0060 1 004 REM00004 Ray Marshall 07-Mar-1983
0061 1 Global edit of all modules. Updated module names, idents,
0062 1 copyright dates. Changed require files to BLISS library.
0063 1
0064 1 !--
```

```

: 66      0064 1 %SBTTL 'Module Level Declarations'
: 67      0065 1
: 68      0066 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
: 69      0197 1
: 70      0198 1
: 71      0199 1 !! TABLE OF CONTENTS:
: 72      0200 1 !!
: 73      0201 1 FORWARD ROUTINE
: 74      0202 1     tstblk : NOVALUE,
: 75      0203 1     tstcnd : NOVALUE,
: 76      0204 1     tstres : NOVALUE,
: 77      0205 1     tsttfe : NOVALUE,
: 78      0206 1     pntbac  : NOVALUE,
: 79      0207 1     negind  : NOVALUE,
: 80      0208 1     remneg  : NOVALUE,
: 81      0209 1     xmarg   : NOVALUE,
: 82      0210 1     remmrg  : NOVALUE;
: 83      0211 1
: 84      0212 1 !!
: 85      0213 1 !! INCLUDE FILES:
: 86      0214 1 !!
: 87      0215 1
: 88      0216 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
: 89      0217 1
: 90      U 0218 1 %IF DSRPLUS %THEN
: 91      U 0219 1 LIBRARY 'REQ:DPLLIB';           ! DSRPLUS BLISS Library
: 92      0220 1 %ELSE
: 93      0221 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
: 94      0222 1 %FI
: 95      0223 1
: 96      0224 1
: 97      0225 1 !! MACROS:
: 98      0226 1 !!
: 99      0227 1
: 100     0228 1 !! EQUATED SYMBOLS:
: 101     0229 1 !!
: 102     0230 1
: 103     0231 1 !! OWN STORAGE:
: 104     0232 1 !!
: 105     0233 1
: 106     0234 1 !! EXTERNAL REFERENCES:
: 107     0235 1 !!
: 108     0236 1
: 109     0237 1 EXTERNAL
: 110     0238 1     gca : gca_definition,
: 111     0239 1     ifstk : ifstack,
: 112     0240 1     savstk : savstack,
: 113     0241 1     irac  : irac_definition;
: 114     0242 1
: 115     0243 1 EXTERNAL
: 116     0244 1     frmstd,
: 117     0245 1     frmstk : form_stack;
: 118     0246 1
: 119     0247 1 EXTERNAL LITERAL                !Error messages
: 120     0248 1     rnfba,
: 121     0249 1     rnflda,
: 122     0250 1     rnfmei,

```

RSDERM  
V04-000

Misc. error checks and messages.  
Module Level Declarations

H 11  
16-Sep-1984 01:40:15  
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]RSDERM.BLI;1

Page 4  
(3)

123	0251	1	rnfmrc,
124	0252	1	rnfmrg,
125	0253	1	rnfmrs,
126	0254	1	rnfnia,
127	0255	1	rnfnic,
128	0256	1	rnfskc,
129	0257	1	rnftfe;
130	0258	1	
131	0259	1	EXTERNAL ROUTINE
132	0260	1	erm,
133	0261	1	ermi,
134	0262	1	ermn,
135	0263	1	lit,
136	0264	1	savres,
137	0265	1	stkfrm;
138	0266	1	
139	0267	1	

RS  
V0

.EXTRN GCA, IFSTK, SAVSTK  
.EXTRN IRAC, FRMSTD, FRMSTK  
.EXTRN RNFBAK, RNFLDE, RNFM EI  
.EXTRN RNFMRC, RNFMRG, RNFMRS  
.EXTRN RNFNIA, RNFNIC, RNFSKC  
.EXTRN RNFTFE, ERM, ERML  
.EXTRN ERMN, LIT, SAVRES  
.EXTRN STKFRM

				.PSECT	\$CODE\$,NOWRT,2	
			000C 00000	.ENTRY	TSTBLK, Save R2,R3	: 0269
53	00000000G	EF	9E 00002	MOVAB	FRMSTD, R3	:
52	00000000G	EF	9E 00009	MOVAB	FRMSTK-20, R2	:
50		63	D0 00010	MOVL	FRMSTD, R0	: 0295
		3B	13 00013	BEQL	2\$	:
50		OF	C4 00015	MULL2	#15, R0	: 0298
6240	04	AC	D1 00018	CMPL	DEPTH, FRMSTK-20[R0]	:
		31	14 0001D	BGTR	2\$	:
		7E	7C 0001F	CLRQ	-(SP)	: 0301
	00000000G	8F	DD 00021	PUSHL	#RNFTFE	:
00000000G	EF	03	FB 00027	CALLS	#3, ERM	:
50		63	D0 0002E 1\$:	MOVL	FRMSTD, R0	: 0304
		1D	15 00031	BLEQ	2\$	:
50		OF	C4 00033	MULL2	#15, R0	:
6240	04	AC	D1 00036	CMPL	DEPTH, FRMSTK-20[R0]	:
		13	14 0003B	BGTR	2\$	:
00000000V	EF	00	FB 0003D	CALLS	#0, PNTBAC	: 0306
	7E	01	CE 00044	MNEGL	#1, -(SP)	: 0307
00000000G	EF	01	FB 00047	CALLS	#1, STKFRM	:
		DE	11 0004E	BRB	1\$	: 0304
		04	00050 2\$:	RET		: 0310

; Routine Size: 81 bytes, Routine Base: \$CODE\$ + 0000

```
185 0311 1 %SBTTL 'TSTCND -- Check and report missing .ENDIFs'
186 0312 1 GLOBAL ROUTINE tstcnd (depth) : NOVALUE =
187 0313 1
188 0314 1 ++
189 0315 1 FUNCTIONAL DESCRIPTION
190 0316 1
191 0317 1
192 0318 1 FORMAL PARAMETERS:
193 0319 1
194 0320 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
195 0321 1
196 0322 1 IMPLICIT INPUTS: None
197 0323 1
198 0324 1 IMPLICIT OUTPUTS: None
199 0325 1
200 0326 1 ROUTINE VALUE:
201 0327 1 COMPLETION CODES: None
202 0328 1
203 0329 1 SIDE EFFECTS: None
204 0330 1
205 0331 1 --
206 0332 1
207 0333 2 BEGIN
208 0334 2 !Check for missing .ENDIF commands. This is temporary coding. In a
209 0335 2 !later version this code should point back to the opening commands,
210 0336 2 !like the code above does for .LIST and .NOTE commands.
211 0337 2
212 0338 2 IF .ifstk [0, ifstk_depth] EQL 0 THEN
213 0339 2 RETURN; !There are no missing .ENDIF commands.
214 0340 2
215 0341 2 IF .depth EQL 0 THEN
216 0342 2 erml (rnfmei) ! Missing .ENDIF commands, and end of file.
217 0343 2 ELSE
218 0344 2 IF .ifstk [.ifstk [0, ifstk_depth], ifstk_req_d] GEQ .depth THEN
219 0345 2 erml (rnfskc); ! Missing .ENDIF commands detected when
220 0346 2 ! unstacking .REQUIRE files
221 0347 2
222 0348 1 END; !End of TSTCND
```

		0000 00000	.ENTRY	TSTCND, Save nothing	: 0312
50	00000000G	EF D0 00002	MOVL	IFSTK, R0	: 0338
		2A 13 00009	BEQL	3\$	: 0341
	04	AC D5 0000B	TSTL	DEPTH	: 0342
		08 12 0000E	BNEQ	1\$	: 0344
	00000000G	8F DD 00010	PUSHL	#RNFMEI	: 0345
		16 11 00016	BRB	2\$	: 0348
50		20 C4 00018 1\$:	MULL2	#32, R0	
	00000000GEF	40 9F 0001B	PUSHAB	IFSTK+16[R0]	
04	AC	9E D1 00022	CMPL	@(SP)+, DEPTH	
		0D 19 00026	BLSS	3\$	
	00000000G	8F DD 00028	PUSHL	#RNFSKC	
00000000G	EF	01 FB 0002E 2\$:	CALLS	#1, ERML	
		04 00035 3\$:	RET		

RSDERM  
V04-000

Misc. error checks and messages.  
TSTCND -- Check and report missing .ENDIFs

L 11  
16-Sep-1984 01:40:15  
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]RSDERM.BLI;1

Page 8  
(5)

; Routine Size: 54 bytes,      Routine Base: \$CODE\$ + 0051

```

224 0349 1 %SBTTL 'TSTRES -- Check for, report, and restore missing .RESTORES'
225 0350 1 ROUTINE tstres (depth) : NOVALUE =
226 0351 1
227 0352 1 ++
228 0353 1 FUNCTIONAL DESCRIPTION
229 0354 1
230 0355 1 This routine checks to see if SAVSTK is ok: i.e., certain open
231 0356 1 .SAVE/SAVE ALL and .RESTORE commands have been closed. If there are any
232 0357 1 unclosed commands it issues an error message and pops the stack.
233 0358 1
234 0359 1 FORMAL PARAMETERS:
235 0360 1
236 0361 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
237 0362 1
238 0363 1 IMPLICIT INPUTS: None
239 0364 1
240 0365 1 IMPLICIT OUTPUTS: None
241 0366 1
242 0367 1 ROUTINE VALUE:
243 0368 1 COMPLETION CODES: None
244 0369 1
245 0370 1 SIDE EFFECTS: None
246 0371 1
247 0372 1 --
248 0373 1
249 0374 2 BEGIN
250 0375 2 OWN
251 0376 2 x;
252 0377 2
253 0378 2 LOCAL
254 0379 2 hold_iseqn,
255 0380 2 hold_page;
256 0381 2
257 0382 2 x = .savstk [0, savstk_depth];
258 0383 2
259 0384 2 IF (.x EQL 0) THEN
260 0385 2 RETURN; ! No unclosed .SAVEs.
261 0386 2
262 0387 2 IF .depth GTR .savstk [.x, savstk_req_d] THEN
263 0388 2 RETURN; ! Unclosed SAVEs but in files still open.
264 0389 2
265 0390 2 !There are unclosed saves to be reported.
266 0391 2
267 0392 2 erml (rnfmr); ! Missing .ReStore commands, at end of file.
268 0393 2
269 0394 2 ! Now point the user back to the point where the corresponding
270 0395 2 ! SAVE/SAVEALL was made.
271 0396 2
272 0397 4 WHILE ( (.savstk [0, savstk_depth] GTR 0)
273 0398 3 AND
274 0399 3 (.depth LEQ .savstk [.savstk [0, savstk_depth] , savstk_req_d]) ) DO
275 0400 3 BEGIN
276 0401 3 hold_page = .irac_ipagen;
277 0402 3 hold_iseqn = .irac_iseqn;
278 0403 3 irac_ipagen = .savstk [.savstk [0, savstk_depth], savstk_ipagen];
279 0404 3 irac_iseqn = .savstk [.savstk [0, savstk_depth], savstk_iseqn];
280 0405 3

```

```

281      0406      3      IF .gca_req_depth NEQ 0 THEN      ! Still in a require file, rewind
282      0407      3      erm (rnfbak      ! stack. Otherwise just quit.
283      0408      3      ,.savstk [.savstk [0, savstk_depth], savstk_fspecc]
284      0409      3      ,.savstk [.savstk [0, savstk_depth], savstk_fspecc] );
285      0410      3
286      0411      3      irac_ipagen = .hold_page;
287      0412      3      irac_iseqn = .hold_iseqn;
288      0413      3      savres (h_restore,-1);      ! Unstack one entry, regardless of identity.
289      0414      2      END;
290      0415      2
291      0416      1      END;      ! End of TSTRES

```

				.PSECT	\$OWNS\$,NOEXE,2		
00000 X:				.BLKB	4		
				.PSECT	\$CODE\$,NOWRT,2		
		007C	00000	TSTRES:	.WORD	Save R2,R3,R4,R5,R6	0350
56	000000000'	EF	9E 00002		MOVAB	X, R6	
55	000000000G	EF	9E 00009		MOVAB	IRAC+12, R5	
54	000000000G	EF	9E 00010		MOVAB	SAVSTK, R4	
66		64	D0 00017		MOVL	SAVSTK, X	0382
50		66	D0 0001A		MOVL	X, R0	0384
		7F	13 0001D		BEQL	3\$	
50		1C	C4 0001F		MULL2	#28, R0	0387
	OC A440	9F	00022		PUSHAB	SAVSTK+12[R0]	
9E	04	AC	D1 00026		CMPL	DEPTH, @ (SP)+	
		72	14 0002A		BGTR	3\$	
	000000000G	8F	DD 0002C		PUSHL	#RNFMR5	0392
000000000G	EF	01	FB 00032		CALLS	#1, ERML	
		64	D0 00039	1\$:	MOVL	SAVSTK, R1	0397
		60	15 0003C		BLEQ	3\$	
50		1C	C5 0003E		MULL3	#28, R1, R0	0399
	OC A440	9F	00042		PUSHAB	SAVSTK+12[R0]	
9E	04	AC	D1 00046		CMPL	DEPTH, @ (SP)+	
		52	14 0004A		BGTR	3\$	
52		65	D0 0004C		MOVL	IRAC+12, HOLD_PAGE	0401
53	FC	A5	D0 0004F		MOVL	IRAC+8, HOLD_ISEQN	0402
50		1C	C5 00053		MULL3	#28, R1, R0	0403
	10 A440	9F	00057		PUSHAB	SAVSTK+16[R0]	
65		9E	D0 0005B		MOVL	@ (SP)+, IRAC+12	
	14 A440	9F	0005E		PUSHAB	SAVSTK+20[R0]	0404
FC	A5	9E	D0 00062		MOVL	@ (SP)+, IRAC+8	
	000000000G	EF	D5 00066		TSTL	GCA+18\$	0406
		19	13 0006C		BEQL	2\$	
	1C A440	9F	0006E		PUSHAB	SAVSTK+28[R0]	0409
		9E	DD 00072		PUSHL	@ (SP)+	
	18 A440	9F	00074		PUSHAB	SAVSTK+24[R0]	0408
		9E	DD 00078		PUSHL	@ (SP)+	
	000000000G	8F	DD 0007A		PUSHL	#RNFBAK	0407
000000000G	EF	03	FB 00080		CALLS	#3, ERM	
	65	52	D0 00087	2\$:	MOVL	HOLD_PAGE, IRAC+12	0411

RSDERM  
V04-000

Misc. error checks and messages.  
TSTRES -- Check for, report, and restore missin

B 12  
16-Sep-1984 01:40:15  
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]RSDERM.BLI;1

Page 11  
(6)

FC	A5	53	DO	0008A	MOVL	HOLD ISEQN, IRAC+8
	7E	01	CE	0008E	MNEGL	#1, =(SP)
	7E	8F	9A	00091	MOVZBL	#175, -(SP)
00000000G	EF	02	FB	00095	CALLS	#2, SAVRES
		9B	11	0009C	BRB	1\$
		04	0009E	3\$:	RET	

: 0412  
: 0413  
: 0397  
: 0416

; Routine Size: 159 bytes, Routine Base: \$CODE\$ + 0087

RSK1  
V04-

```

293 0417 1 GLOBAL ROUTINE tsttfe (depth) : NOVALUE =
294 0418 1
295 0419 1 ++
296 0420 1 FUNCTIONAL DESCRIPTION
297 0421 1
298 0422 1 This routine does nothing more than merely pass control to each of
299 0423 1 the respective test routines to check for unclosed lists, saves, etc.
300 0424 1 Oh, one exception: It now checks for being within a .LITERAL. If it
301 0425 1 finds this to be true, it issues an error and closes the literal.
302 0426 1
303 0427 1 FORMAL PARAMETERS:
304 0428 1
305 0429 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
306 0430 1
307 0431 1 IMPLICIT INPUTS: None
308 0432 1
309 0433 1 IMPLICIT OUTPUTS: None
310 0434 1
311 0435 1 ROUTINE VALUE:
312 0436 1 COMPLETION CODES: None
313 0437 1
314 0438 1 SIDE EFFECTS: None
315 0439 1
316 0440 1 --
317 0441 1
318 0442 1 BEGIN
319 0443 2
320 0444 2 IF .gca literal THEN ! If within a literal;
321 0445 3 BEGIN
322 0446 3 erm (rnflde ! say that it wasn't closed,
323 0447 3 ,CH$PTR(UPLIT('At or near end of document'))! and where
324 0448 3 ,26);
325 0449 3 lit( h_end_literal ) ! and then close it.
326 0450 2 END;
327 0451 2 tstblk (.depth); ! Check for unclosed lists, notes, and literals.
328 0452 2 tstcnd (.depth); ! Check for unclosed ifs and ifnots.
329 0453 2 tstres (.depth); ! Check for unclosed saves.
330 0454 2
331 0455 1 END; ! End of TSTTFE

```

```

20 64 6E 65 20 72 61 65 6E 20 72 6F 20 74 41 0000 P.AAA: .PSECT $SPLITS,NOWRT,NOEXE,2
00 00 74 6E 65 6D 75 63 6F 64 20 66 6F 0000F .ASCII \At or near end of document\<0><0>

```

```

0004 00000 .PSECT $CODE$,NOWRT,2
52 FED4 CF 9E 00002 .ENTRY TSTTFE, Save R2 : 0417
1E 00000000G EF E9 00007 MOVAB TSTBLK, R2 : 0444
00000000' 1A DD 0000E BLBC GCA+80, 1$ : 0446
00000000G 8F DD 00016 PUSHL #26 : 0447
PUSHAB P.AAA : 0446
PUSHL #RNFLDE

```

RSDERM  
V04-000

Misc. error checks and messages.

TSTRES -- Check for, report, and restore missin

D 12

16-Sep-1984 01:40:15

14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742

[RUNOFF.SRC]RSDERM.BLI;1

Page 13

(7)

00000000G	EF		03	FB	0001C	CALLS	#3, ERM
			3C	DD	00023	PUSHL	#60
00000000G	EF		01	FB	00025	CALLS	#1, LIT
		04	AC	DD	0002C	PUSHL	DEPTH
	62		01	FB	0002F	CALLS	#1, TSTBLK
		04	AC	DD	00032	PUSHL	DEPTH
51	A2		01	FB	00035	CALLS	#1, TSTCND
		04	AC	DD	00039	PUSHL	DEPTH
0087	C2		01	FB	0003C	CALLS	#1, TSTRES
			04	00041		RET	

: 0449  
: 0451  
: 0452  
: 0453  
: 0455

; Routine Size: 66 bytes, Routine Base: \$CODE\$ + 0126

; 332 0456 1

```
334 0457 1 GLOBAL ROUTINE pntbac : NOVALUE =
335 0458 1 ++
336 0459 1 FUNCTIONAL DESCRIPTION
337 0460 1
338 0461 1 This routine saves the input file line/page information. It then
339 0462 1 substitutes information from FRMSTK for that information. Then ERM is
340 0463 1 called to output a message indicating where to look in the input file
341 0464 1 for the command that made the stack entry. But ERM does not have
342 0465 1 enough flexibility to accept all that information as parameters.
343 0466 1 However, it can pick it up from IRAC, which it does. In other words,
344 0467 1 this routine fakes out ERM a bit.
345 0468 1
346 0469 1 After ERM returns, the original information is restored.
347 0470 1
348 0471 1 --
349 0472 2 BEGIN
350 0473 2
351 0474 2 LOCAL
352 0475 2 hold_iseqn,
353 0476 2 hold_page;
354 0477 2
355 0478 2 hold_page = .irac_ipagen;
356 0479 2 hold_iseqn = .irac_iseqn;
357 0480 2 irac_ipagen = .frmstk [.frmstd, frmstk_ipagen];
358 0481 2 irac_iseqn = .frmstk [.frmstd, frmstk_iseqn];
359 0482 2 erm {rnfbak
360 0483 2     .frmstk [.frmstd, frmstk_fspecc]
361 0484 2     .frmstk [.frmstd, frmstk_fspecc] );
362 0485 2 irac_ipagen = .hold_page;
363 0486 2 irac_iseqn = .hold_iseqn;
364 0487 1 END;
```

!End of PNTBAC

```
003C 00000
55 00000000G FF 9E 00002
54 00000000G EF 9E 00009
52 64 D0 00010
53 FC A4 D0 00013
50 00000000G EF 0F C5 00017
64 FC A4 FC A540 D0 0001F
FC A4 08 A540 DD 00023
04 A540 DD 00029
00000000G 8F DD 00031
00000000G EF 03 FB 00037
64 FC A4 52 D0 0003E
FC A4 53 D0 00041
04 00045
```

```
.ENTRY PNTBAC, Save R2,R3,R4,R5
MOVAB FRMSTK-32, R5
MOVAB IRAC+12, R4
MOVL IRAC+12, HOLD_PAGE
MOVL IRAC+8, HOLD_ISEQN
MULL3 #15, FRMSTD, R0
MOVL FRMSTK-32[R0], IRAC+12
MOVL FRMSTK-36[R0], IRAC+8
PUSHL FRMSTK-24[R0]
PUSHL FRMSTK-28[R0]
PUSHL #RNFBK
CALLS #3, ERM
MOVL HOLD_PAGE, IRAC+12
MOVL HOLD_ISEQN, IRAC+8
RET
```

```
: 0457
:
:
: 0478
: 0479
: 0480
:
: 0481
: 0484
: 0483
: 0482
:
: 0485
: 0486
: 0487
```

; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 0168

; 365 0488 1

```

: 367 0489 1 GLOBAL ROUTINE negind : NOVALUE =
: 368 0490 1
: 369 0491 1 ++
: 370 0492 1 FUNCTIONAL DESCRIPTION:
: 371 0493 1
: 372 0494 1     NEGIND keeps track of attempted indents past the left side of the page.
: 373 0495 1     To avoid issuing so many error messages that the user would be annoyed,
: 374 0496 1     it only issues an error message for the first such attempt. Other
: 375 0497 1     violations are simply counted; the summary count is output by REMNEG
: 376 0498 1     later.
: 377 0499 1
: 378 0500 1 FORMAL PARAMETERS:      None
: 379 0501 1
: 380 0502 1 IMPLICIT INPUTS      None
: 381 0503 1
: 382 0504 1 IMPLICIT OUTPUTS     None
: 383 0505 1
: 384 0506 1 ROUTINE VALUE:
: 385 0507 1 COMPLETION CODES:      None
: 386 0508 1
: 387 0509 1 SIDE EFFECTS:          None
: 388 0510 1
: 389 0511 1 --
: 390 0512 1
: 391 0513 2 BEGIN
: 392 0514 2
: 393 0515 2 IF .gca_nia EQL 0 THEN
: 394 0516 2     erml(rnfnia);
: 395 0517 2
: 396 0518 2 gca_nia = .gca_nia + 1;
: 397 0519 1 END;

```

!End of NEGIND

		0004 00000	.ENTRY	NEGIND, Save R2	: 0489
52	00000000G	EF 9E 00002	MOVAB	GCA+160, R2	: 0515
		62 D5 00009	TSTL	GCA+160	: 0516
		0D 12 0000B	BNEQ	1\$	: 0518
		8F DD 0000D	PUSHL	#RNFNIA	: 0519
00000000G	EF	01 FB 00013	CALLS	#1, ERML	
		62 D6 0001A 1\$:	INCL	GCA+160	
		04 0001C	RET		

; Routine Size: 29 bytes, Routine Base: \$CODE\$ + 01AE

; 398 0520 1

```

: 400 0521 1 GLOBAL ROUTINE remneg : NOVALUE =
: 401 0522 1
: 402 0523 1 ++
: 403 0524 1 FUNCTIONAL DESCRIPTION:
: 404 0525 1
: 405 0526 1 Issues an error message indicating how many negative
: 406 0527 1 indents have been detected. Then resets the count to zero.
: 407 0528 1
: 408 0529 1 Used together with NEGIND.
: 409 0530 1
: 410 0531 1 FORMAL PARAMETERS: None
: 411 0532 1
: 412 0533 1 IMPLICIT INPUTS: None
: 413 0534 1
: 414 0535 1 IMPLICIT OUTPUTS: None
: 415 0536 1
: 416 0537 1 ROUTINE VALUE:
: 417 0538 1 COMPLETION CODES: None
: 418 0539 1
: 419 0540 1 SIDE EFFECTS: None
: 420 0541 1
: 421 0542 1 --
: 422 0543 1
: 423 0544 2 BEGIN
: 424 0545 2
: 425 0546 2 IF .GCA_NIA GTR 1 THEN ! 1 or zero means all accounted for.
: 426 0547 2 erm (rmfnc, .gca_nia - 1); ! Subtract off reported negative indent
: 427 0548 2
: 428 0549 2 gca_nia = 0; ! This starts the counting over again.
: 429 0550 1 END; ! End of REMNEG

```

```

                                0004 00000
                                52 00000000G EF 9E 00002
                                01 62 D1 00009
                                11 15 0000C
                                7E 62 01 C3 0000E
                                00000000G 8F DD 00012
                                EF 02 FB 00018
                                62 D4 0001F 1$:
                                04 00021
                                .ENTRY REMNEG, Save R2
                                MOVAB GCA+160, R2
                                CMPL GCA+160, #1
                                BLEQ 1$
                                SUBL3 #1, GCA+160, -(SP)
                                PUSHL #RMFNC
                                CALLS #2, ERMN
                                CLRL GCA+160
                                RET
: 0521
: 0546
: 0547
: 0549
: 0550

```

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 01CB

; 430 0551 1

```

: 432 0552 1 %SBTTL 'XMARG -- Controls R/L margin violation error messages'
: 433 0553 1 GLOBAL ROUTINE xmarg : NOVALUE =
: 434 0554 1
: 435 0555 1 ++
: 436 0556 1 FUNCTIONAL DESCRIPTION:
: 437 0557 1
: 438 0558 1 XMARG keeps track of attempted margin crossings (i.e., left
: 439 0559 1 margin exceeds right margin). To avoid issuing so many
: 440 0560 1 messages that the user would be annoyed, it only issues
: 441 0561 1 an error message for the first such attempt. Other violations
: 442 0562 1 are simply counted; the summary count is output by REMMRG later.
: 443 0563 1
: 444 0564 1 FORMAL PARAMETERS: None
: 445 0565 1
: 446 0566 1 IMPLICIT INPUTS: None
: 447 0567 1
: 448 0568 1 IMPLICIT OUTPUTS: None
: 449 0569 1
: 450 0570 1 ROUTINE VALUE:
: 451 0571 1 COMPLETION CODES: None
: 452 0572 1
: 453 0573 1 SIDE EFFECTS: None
: 454 0574 1
: 455 0575 1 --
: 456 0576 1
: 457 0577 2 BEGIN
: 458 0578 2
: 459 0579 2 IF .gca_xmarg EQL 0 THEN
: 460 0580 2 erml(rnfmrg);
: 461 0581 2
: 462 0582 2 gca_xmarg = .gca_xmarg + 1;
: 463 0583 1 END;

```

!End of XMARG

52	00000000G	EF	9E	00002	.ENTRY	XMARG, Save R2	: 0553
62		D5	00009		MOVAB	GCA+164, R2	: 0579
0D		12	0000B		TSTL	GCA+164	: 0580
8F		DD	0000D		BNEQ	1\$	: 0582
01		FB	00013		PUSHL	#RNFMRG	: 0583
62		D6	0001A	1\$:	CALLS	#1, ERML	
04		0001C			INCL	GCA+164	
					RET		

; Routine Size: 29 bytes, Routine Base: \$CODE\$ + 01ED

; 464 0584 1

```

: 466 0585 1 %SBTTL 'REMMRG -- reports # of crossed margin errors'
: 467 0586 1 GLOBAL ROUTINE REMMRG : NOVALUE =
: 468 0587 1
: 469 0588 1 ++
: 470 0589 1 FUNCTIONAL DESCRIPTION:
: 471 0590 1
: 472 0591 1 Issues an error message indicating how many crossed
: 473 0592 1 margins have been detected. Then resets the count to zero.
: 474 0593 1
: 475 0594 1 Used together with XMARG.
: 476 0595 1
: 477 0596 1 FORMAL PARAMETERS: None
: 478 0597 1
: 479 0598 1 IMPLICIT INPUTS: None
: 480 0599 1
: 481 0600 1 IMPLICIT OUTPUTS: None
: 482 0601 1
: 483 0602 1 ROUTINE VALUE:
: 484 0603 1 COMPLETION CODES: None
: 485 0604 1
: 486 0605 1 SIDE EFFECTS: None
: 487 0606 1
: 488 0607 1 --
: 489 0608 1
: 490 0609 2 BEGIN
: 491 0610 2
: 492 0611 2 IF .gca_xmarg GTR 1 THEN ! 1 or zero means all accounted for.
: 493 0612 2 ermnr(rnfmrc, .gca_xmarg - 1); ! Subtract off reported negative indent
: 494 0613 2
: 495 0614 2 gca_xmarg = 0; ! This starts the counting over again.
: 496 0615 1 END; ! End of REMMRG

```

```

                                0004 00000
                                EF 9E 00002
                                01 62 D1 00009
                                11 15 0000C
7E                                01 C3 0000E
                                8F DD 00012
                                00000000G EF 02 FB 00018
                                62 D4 0001F 1$:
                                04 00021
ENTRY REMMRG, Save R2
MOVAB GCA+164, R2
CMPL GCA+164, #1
BLEQ 1$
SUBL3 #1, GCA+164, -(SP)
PUSHL #RNFMRC
CALLS #2, ERMN
CLRL GCA+164
RET

```

; Routine Size: 34 bytes, Routine Base: \$CODE\$ + 020A

```

: 497 0616 1
: 498 0617 1 END !End of module
: 499 0618 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	556	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$OWN\$	4	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$SPLIT\$	28	NOVEC,NOWRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	28	2	86	00:00.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RSDERM/OBJ=OBJ\$:RSDERM MSRC\$:RSDERM/UPDATE=(ENH\$:RSDERM)

: Size: 556 code + 32 data bytes  
: Run Time: 00:10.7  
: Elapsed Time: 00:28.5  
: Lines/CPU Min: 3478  
: Lexemes/CPU-Min: 15303  
: Memory Used: 58 pages  
: Compilation Complete

0348 AH-BT13A-SE  
VAX/VMS V4.0

# DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

0348	0349	0350	0351	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399								
0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459
0460	0461	0462	0463	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511	0512	0513	0514	0515	0516	0517	0518	0519
0520	0521	0522	0523	0524	0525	0526	0527	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575	0576	0577	0578	0579
0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699
0700	0701	0702	0703	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751	0752	0753	0754	0755	0756	0757	0758	0759
0760	0761	0762	0763	0764	0765	0766	0767	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815	0816	0817	0818	0819
0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939
0940	0941	0942	0943	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991	0992	0993	0994	0995	0996	0997	0998	0999
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059
1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179
1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239
1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299
1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419
1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479
1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539
1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659
1660	1661	1662	1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695	1696	1697	1698	1699</																				